โปรแกรมเปลี่ยนฐานเลขระหว่างฐาน 2 ถึง 16

วัชระ รอดสัมฤทธิ์

อันที่จริงเราสามารถหาโปรแกรมเปลี่ยนฐานเลขที่เขียนสำเร็จรูปไว้แล้วได้ง่าย แต่โดยทั่วไปมักจะเป็นโปรแกรมที่เขียนขึ้นมาเพื่อเปลี่ยนฐานเลขระหว่างฐานใดฐานหนึ่งเท่านั้น ในภาษา MBASIC ก็มีคำสั่งเปลี่ยนฐานเลขอยู่สองคำสั่งคือ OCT$(X) และ HEX$(X) ซึ่งเป็นการเปลี่ยนค่า X ในเลขฐานสิบให้เป็นเลขฐานแปด และฐานสิบหกตามลำดับ แต่โปรแกรมที่จะนำเสนอต่อไปนี้ เป็นโปรแกรมที่เขียนขึ้นเพื่อใช้เปลี่ยนฐานเลขใดๆ ซึ่งอยู่ระหว่างฐาน 2 ถึง 16 ผู้ใช้สามารถดัดแปลงให้เปลี่ยนค่าตัวเลขให้ได้มากกว่าฐาน 16 ก็ได้ ภาษาที่ใช้เขียนเป็นภาษา MBASIC ที่ใช้ในเครื่องไอบีเอ็มพีซี

หลักการ

โปรแกรมนี้เขียนขึ้นโดยอาศัยหลักการดังนี้ "เมื่อ X เป็นจำนวนเลขในฐานใดๆ (สมมติให้เป็นฐาน BASE1) ต้องการเปลี่ยนค่า X ให้เป็นเลขในอีกฐานหนึ่ง (สมมติให้เป็นฐาน BASE2) ทำได้โดยการเปลี่ยนค่า X ในฐาน BASE1 ให้เป็นเลขในฐานสิบ แล้วจึงเปลี่ยนเลขฐานสิบนี้ให้เป็นเลขฐาน BASE2 อีกทีหนึ่ง"

การเปลี่ยนเลขฐานใดๆ (ฐาน BASE1) ให้เป็นเลขฐานสิบ (DEC, Decimal base) หาได้จากสูตรที่ 1

$$DEC = \sum_{N=1}^{L} V_N(BASE1)^{L-N}$$

เมื่อ L คือจำนวนหลักของตัวเลข

V_N คือตัวเลขในหลักที่ N

BASE1 คือฐานเลขที่เราต้องการเปลี่ยน

เช่นต้องการหาว่า (2017) ฐาน 8 มีรูปอย่างไรในฐานสิบก็ทำได้ดังนี้

(2017) ฐาน 8 มีอยู่ 4 หลัก (L = 4) หลักที่ 1 (V_1) คือ 2 หลักที่ 2 (V_2) คือ 0 หลักที่ 3 (V_3) คือ 1 และหลักที่ 4 (V_4) คือ 7 BASE1 คือ 8 (ฐาน 8) แทนค่าต่างๆ ลงในสูตรที่ (1) จะได้

$$DEC = \sum_{N=1}^{4} V_N(8)^{4-N}$$

$$= V_1(8)^{4-1} + V_2(8)^{4-2} + V_3(8)^{4-3} + V_4(8)^{4-4}$$

แทนค่า V_1, V_2, V_3 และ V_4

$$= 2 \times 8^{4-1} + 0 \times 8^{4-2} + 1 \times 8^{4-3} + 7 \times 8^{4-4}$$

$$= 1024 + 0 + 8 + 7$$

$$= (1039) \text{ ฐานสิบ}$$

การเปลี่ยนฐานเลขจากฐานใด ๆ ให้เป็นฐานสิบในโปรแกรมนี้ จะใช้สูตรที่ (1) คำนวณ

เมื่อต้องการเปลี่ยนเลขฐานสิบให้เป็นเลขฐานอื่น (ฐาน BASE2) ทำได้ดังสูตรที่ 2 ดังนี้

ให้นำเลขฐานสิบเป็นตัวตั้งหารด้วย BASE2 หารได้เท่าไรเก็บเศษไว้ นำผลหารที่ได้เป็นตัวตั้งแล้วหารด้วย BASE2 หารได้เท่าไรเก็บเศษไว้อีก ทำเช่นนี้จนกว่าผลลัพธ์ของการหารจะมีค่าเป็นศูนย์ แล้วนำเศษทั้งหมดมาเขียนเรียงกันโดยเริ่มต้นตั้งแต่เศษตัวสุดท้ายจนถึงเศษตัวแรก

ต้องการเปลี่ยน (1039) ฐานสิบ ให้เป็นเลขฐาน 16 (BASE2 = 16) โดยอาศัยวิธีการในสูตรที่ 2 จะได้เป็น

1039 ÷ 16 = 64 เศษ 15

64 ÷ 16 = 4 เศษ 0

4 ÷ 16 = 0 เศษ 4

เรียงลำดับเศษจากตัวสุดท้ายไปหาตัวแรกจะได้ (4) (0) (15) ในเลขฐาน 16 เราแทน "15" ด้วยตัวอักษร F ดังนั้น

(1039) ฐานสิบ = (40F) ฐาน 16

นำสูตรที่ (1) และ (2) มาเขียนโปรแกรม เมื่อเขียนเสร็จแล้วจะได้ดังรูปที่ 1

อธิบายโปรแกรม

เพื่อความสะดวกในการศึกษาและแก้ไขดัดแปลงโปรแกรม จึงได้อธิบายโปรแกรมแต่ละบรรทัดดังรูปที่ 2

```
1 '                This program written by
2 '                  Mr.Wachara Rodsumrid
3 '                      18 May 86
4 '
5 '
6 CLS:KEY OFF:DIM DIGIT$(16)
7 FOR I = 0 TO 15 :READ DIGIT$(I):NEXT I
8 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
9 '
10 REM  ........print heading.................
11 LOCATE 6,10:PRINT STRING$(60,"=")
12 LOCATE 4,28:PRINT "Numeric Base Conversion"
13 REM .........INPUT BASE TO CONVERT.......
14 LOCATE 8,12:PRINT "To convert from base........."
15 LOCATE 9,13:PRINT "(print END for exit.)"
16 LOCATE 8,40:PRINT STRING$(15," "):LOCATE 8,40:LINE INPUT BASE1$
17 BASE1#=VAL(BASE1$)
18 IF BASE1$="END" OR  BASE1$="end" THEN CLS :LOCATE 12,25:PRINT "*******GOOD LU
CK********":KEY ON:END
19 REM ..........check error ..........................
20 LL=0
21 FOR T=1 TO LEN(BASE1$)
22 B$=MID$(BASE1$,T,1)
23 IF B$<"0" OR B$>"9" THEN BEEP:GOSUB 71:GOTO 14
24 NEXT T
25 IF BASE1#<2 OR BASE1#>16 OR LEN(BASE1$)>2 THEN BEEP:GOSUB 74:GOTO 14
26 LL=LL+3
27 REM ..........input new base............
28 LOCATE 11,12:PRINT "to new output base.............."
29 LOCATE 12,13:PRINT "( 2 thru 16) "
30 LOCATE 11,40:PRINT STRING$(20," "):LOCATE 11,40:LINE INPUT BASE2$
31 BASE2#=VAL(BASE2$)
32 FOR T=1 TO LEN(BASE2$)
33 B$=MID$(BASE2$,T,1)
34 IF B$<"0" OR B$>"9" THEN BEEP:GOSUB 71:GOTO 28
35 NEXT T
36 IF BASE2#<2 OR BASE2#>16 OR LEN(BASE2$)>2 THEN BEEP:GOSUB 74:GOTO 28
37 REM .........input value.....................
38 LOCATE 14,12:PRINT "Your value in base ";BASE1$;" ?.......":LOCATE 14,40:PRIN
T STRING$(40," "):LOCATE 14,40:LINE INPUT X$
39 L=LEN(X$)
40 REM ..........  check value.......................
41 IF L=0 THEN 36
42 FOR I=1 TO L
43 V$=MID$(X$,I,1)
44 IF BASE1#=<10 AND (ASC(V$)<48 OR ASC(V$)>57) THEN BEEP:GOSUB 77:GOTO 38
45 IF VAL(V$)=>BASE1# THEN BEEP:GOSUB 77:GOTO 38
46 IF BASE1#>10 THEN IF ASC(V$)<48 OR (ASC(V$)>57 AND ASC(V$)<65) OR ASC(V$)>(54
+BASE1#) THEN BEEP:GOSUB 77:GOTO 38
47 NEXT I
48 REM ..............convert to decimal base.................
49 DEC#=0
50 FOR N= 1 TO L:FOR J=0 TO (BASE1#-1)
51 IF DIGIT$(J)<>MID$(X$,N,1) THEN 53
52 DEC#=DEC#+(J*BASE1#^(L-N))
53 NEXT J
54 NEXT N
55 LOCATE 16,10:PRINT STRING$(60,"=")
56 REM LOCATE 18,20 :PRINT  X$;" in base ";BASE1#;" = ";DEC#;" in  decimal base.
"
57 IF DEC#>28999999999# THEN BEEP:LOCATE 19,20 :COLOR 0,7:PRINT "This value is t
oo large for me to calculate.":COLOR 7,0:GOTO 67
58 REM .........convert  dec to new base...................
59 RESULT$=""
60 WHILE DEC#<>0
61 REMAINDER=INT(((DEC#/BASE2#)-INT(DEC#/BASE2#))*BASE2#+.5)
62 RESULT$=DIGIT$(REMAINDER)+RESULT$
63 DEC#=INT(DEC#/BASE2#)
64 WEND
65 IF RESULT$="" THEN RESULT$="0"
66 LOCATE 19,20:PRINT X$;" in base";BASE1#;" =  ";RESULT$;"  in base ";BASE2#
67 PRINT :PRINT:PRINT "      Press <SPACEBAR> to another value....."
68 K$=INKEY$:IF K$<>" " THEN 68 ELSE CLS:GOTO 10
69 '..............Delay time.....................
70  FOR DELAY= 1 TO 1500: NEXT DELAY:RETURN
71 '............Check letters,space and signs..................
72 LOCATE (9+LL),12:COLOR 0,7:PRINT "letters,space and any signs are not allowed
....":COLOR 7,0:GOSUB 69:LOCATE(9+LL),12:PRINT STRING$(50," ")
73 RETURN
74 '.......check base.............
75 LOCATE (9+LL),12:COLOR 0,7:PRINT "base 2 thru base 16 only !!":GOSUB 69: COLO
R 7,0:LOCATE (9+LL),12:PRINT STRING$(35," ")
76 RETURN
77 ' ............print value error.............
78 LOCATE 14,50:COLOR 0,7:PRINT "Invalid Value ? ":COLOR 7,0: GOSUB 69:LOCATE 14
,50:PRINT STRING$(20," "):RETURN
```

รูปที่ 1 โปรแกรมเปลี่ยนฐานเลข

รูปที่ 2 ตารางอธิบายรายละเอียดของโปรแกรม

บรรทัดที่	คำอธิบาย
1–5	เป็นหมายเหตุของผู้เขียนโปรแกรม
6–8	ล้างจอภาพและกำหนดตัวเลขสำหรับใช้ ตั้งแต่ฐานสองถึงฐานสิบหก มีอยู่ 15 ตัว คือ 0, 1, 2,.....ถึง F เก็บไว้ในตัวแปรสตริงก์ที่ชื่อว่า DIGIT$(I) ซึ่งเป็นตัวแปรแบบอะเรย์ใน 1 มิติ I มีค่าตั้งแต่ 0 ถึง 15 ค่า I ซึ่งเป็นค่ากำหนดของตัวแปรจะใช้ประโยชน์ในการคำนวณและเรียงลำดับตัวเลข เช่น DIGIT$(12) = "C" เป็นตัวกำหนดว่า C มีค่าเท่ากับ 12
10–12	เป็นการพิมพ์ชื่อเรื่อง
13–18	ป้อนฐานเลขที่ต้องการเปลี่ยนเก็บไว้ในตัวแปร BASE1$ ซึ่งมีค่าอยู่ระหว่าง 2 ถึง 16 ถ้าต้องการออกจากโปรแกรมให้พิมพ์คำว่า "END"
19–26	ตรวจสอบดูว่า ผู้ใช้ใส่ฐานเลขน้อยกว่าฐานสองหรือมากกว่าฐาน 16 หรือไม่ และถ้าใส่ตัวอักษรหรือช่องว่าง เครื่องจะให้ผู้ใช้ป้อนฐานเลขอีกครั้ง
27–36	ป้อนฐานเลขฐานใหม่ (BASE2$) และตรวจสอบความถูกต้องของ BASE2$ เช่นเดียวกับหมายเลขบรรทัดที่ 19–26
37–39	ป้อนจำนวนที่เราต้องการเปลี่ยนฐาน ซึ่งเป็นเลขในฐาน BASE1$ เก็บค่านี้ไว้ในตัวแปร X$
40–47	ตรวจสอบค่า X$ ว่าเป็นเลขฐาน BASE1 จริงหรือไม่ ถ้าไม่ใช่จะพิมพ์คำว่า "Invalid Value?" พร้อมกับให้ป้อนค่า X$ ใหม่
48–55	เปลี่ยนค่า X$ เป็นเลขฐานสิบโดยใช้สูตรที่ 1 ให้ DEC≠ เป็นตัวแปรสำหรับเก็บเลขฐานสิบที่คำนวณได้นี้
56	เป็นการพิมพ์ค่า DEX≠ บรรทัดนี้ใส่ REM ไว้จึงไม่แสดงผลบนจอภาพ เขียนไว้สำหรับใช้ประโยชน์เวลาตรวจหาข้อผิดพลาดของโปรแกรม
57	เป็นบรรทัดที่บอกขีดจำกัดของโปรแกรมนี้ ค่า X$ ในเลขฐานใดๆ เมื่อแปลงเป็นเลขฐานสิบแล้วจะต้องมีค่าไม่เกิน 28,999,999,999 ถ้าเกินนี้แล้วจะได้ค่าคลาดเคลื่อน (เครื่องที่ผู้เขียนใช้เป็นแบบไอบีเอ็มคอมแพต)
58–64	เปลี่ยนตัวเลขฐานสิบ (DEC≠) ให้เป็นเลขฐานที่ต้องการโดยใช้สูตรที่ 2 บรรทัดที่ 61–62 เป็นการเก็บเศษที่ได้จากการหาร(REMIANDER) แล้วนำเศษไปเรียงเก็บไว้ในตัวแปรที่ชื่อว่า RESVLT$ ทำเช่นนี้จนผลหารมีค่าเป็นศูนย์ จึงหลุดออกจาก While-Wend loop ผู้เขียนลองเปลี่ยนคำสั่งในบรรทัดที่ 61 โดยใช้คำสั่ง MOD และบรรทัดที่ 63 ใช้คำสั่ง \ (Integer devision) พบว่าเกิดการ overflow เมื่อค่า DEC≠ มากกว่า 32767 จึงตกลงที่จะใช้คำสั่ง INT ดีกว่า ถึงแม้จะดูซับซ้อนกว่าก็ตาม ในบรรทัดที่ 61 จะเห็นว่ามี 0.5 บวกอยู่ตอนท้าย ใส่ไว้เพื่อเป็นการปัดจุดทศนิยม ทำให้เราได้เศษที่มีค่าถูกต้องเพราะคำสั่ง INT ไม่ได้มีการปัดจุดทศนิยมให้เรา
65	เพราะเรากำหนด RESULT$ ในบรรทัดที่ 59 ให้เป็น null string เมื่อเราป้อนค่า X$ = "0" ผลการคำนวณจะได้เลขฐาน 10 (DEC≠) เป็นศูนย์ เมื่อเป็นเช่นนี้ จึงไม่มีการทำงานใน While-Wend loop ผลลัพธ์สุดท้ายก็จะกลายเป็น null string เช่นเดิม จึงต้องใส่บรรทัดนี้กันไว้
66	เป็นการพิมพ์ผลลัพธ์ที่ได้บนจอภาพ
67–68	กด spacebar เมื่อต้องการหาค่าจำนวนต่อไป เครื่องจะกลับไปเริ่มทำงานบรรทัดที่ 10 ใหม่
69–70	เป็นการหน่วงเวลาการทำงาน เพื่อให้ผู้ใช้อ่านข้อความบนจอภาพได้ทัน
71–76	แสดงข้อความผิดพลาดขณะที่ป้อนฐาน BASE1$ และ BASE2$ เนื่องจากข้อความที่แสดงมีใจความเหมือนกัน จึงนำมาเขียนไว้ในโปรแกรมย่อยเดียวกัน และใช้ตัวแปร LL เป็นตัวกำหนดบรรทัดที่แสดงข้อความ
77–78	แสดงข้อความผิดพลาดสำหรับค่า X$

รูปที่ 3 ตารางการดัดแปลงโปรแกรมสำหรับแอปเปิ้ลซอฟท์

คำสั่งที่ต้องเปลี่ยนในโปรแกรมนี้	คำสั่งเมื่อใช้กับเครื่องแอปเปิ้ล
1. CLS	HOME
2. KEY OFF และ KEY ON	ตัดทิ้งได้เลย
3. PRINT STRING$ (N, "=") เมื่อ N เป็นตัวเลข	FOR I = 1 TO N:? "=";: NEXT I
4. PRINT STRING$ (N, " ")	CALL-958
5. LOCATE (row, column)	VTAB(row):HTAB(column) เปลี่ยนตัวเลขของคอลัมน์ใหม่ให้เหมาะสมกับจอขนาด 40 คอลัมน์
6. BEEP	PRINT CHR$(7)
7. WHILE-WEND LOOP ในบรรทัดที่ 60–64	60 IF DEC## = 0 THEN 65 61, 62, 63 เหมือนเดิม 64 GOTO 60
8. COLOR 0, 7	INVERSE
9. COLOR 7, 0	NORMAL
10. LINE INPUT (ตัวแปร)	INPUT " "; (ตัวแปร)
11. K$ = INKEY$	GET K$
12. เครื่องหมาย ## ที่อยู่ท้ายตัวแปรตัวเลขเป็นตัวระบุว่าเป็นจำนวนเลขแบบ double precision	ตัดเครื่องหมาย ## ทิ้งได้
13. บรรทัดที่ 57	ต้องเปลี่ยนขีดจำกัดให้เหมาะสมกับเครื่อง

การดัดแปลงโปรแกรมไปใช้ในแอปเปิ้ลซอฟท์เบสิก

สำหรับผู้ที่ใช้เครื่องแอปเปิ้ล การเปลี่ยนคำสั่งบางคำสั่งสามารถนำโปรแกรมไปใช้ได้เลยดังรูปที่ 3

ข้อเสนอแนะ

เมื่อต้องการจะขยายโปรแกรม ให้ใช้ในการเปลี่ยนฐานเลขให้ได้มากกว่าฐาน 16 ทำได้โดยกำหนดตัวแปร DIGIT$(I) ในบรรทัดที่ 6–8 เสียใหม่ และเปลี่ยนค่าต่าง ๆ สำหรับใช้ในการตรวจสอบฐานเลขในบรรทัดที่ 19–26 และในบรรทัดที่ 27–36 ให้สอดคล้องกับฐานเลขที่เราขยายออกไป

โปรแกรมนี้ยังมีขีดจำกัดก็คือ เปลี่ยนฐานเลขได้เฉพาะตัวเลขที่เป็นจำนวนเต็ม ถ้าขยายความสามารถเปลี่ยนฐานเลขไปถึงจำนวนที่มีจุดทศนิยมก็จะทำให้โปรแกรมสมบูรณ์ยิ่งขึ้น

หนังสืออ้างอิง

1. Graham, J. Lyle, **Your IBM PC**, McGraw-Hill, 1983, p579.
2. สมนึก ศรีวิศาล, **การเปลี่ยนฐานเลข**, วารสารไมโครคอมพิวเตอร์ ฉบับที่ 7 หน้า 124–129.